THINKING · LEARNING · CREATING

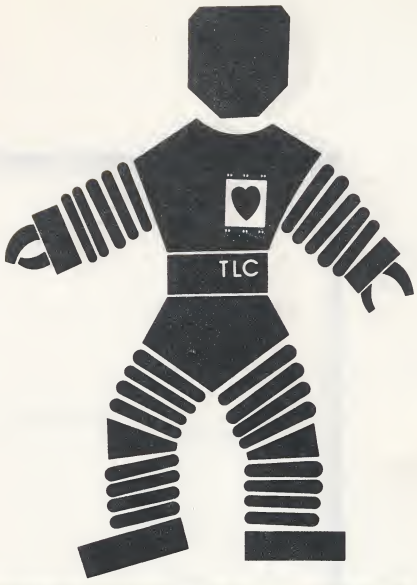
TLC

ATARI®

for Growing Minds

Volume 2

® A registered trademark of Arari, Inc.



THINKING · LEARNING · CREATING

TLC

ATARI®

for Growing Minds

Volume 2

by Nancy Taitt

CREATIVE LEARNING ASSOCIATION, INC.
RR#4, Box 330
Charleston, Illinois 61920

ISBN 0-88193-072-5

© 1983 CREATIVE LEARNING ASSOCIATION, INC.

Charleston, Illinois 61920

All rights reserved. No part of this book, except for brief passages in articles and reviews that refer to both author and publisher, may be reproduced without written permission of the publisher.

ATARI TABLE OF CONTENTS

CHAPTER 1: INPUT IS OFTEN BETTER THAN OUTPUT Page

INPUT. 1
String Variables 3
DIM X$ 4
for Reverse Printing. 9
CTRL 1 9
Review 12

CHAPTER 2: UP, UP, AND ARRAY!

Arrays 13
DIM X. 13
REM. 16
Review 18

CHAPTER 3: FOR YOUR NEXT PROGRAM, TRY THIS!

FOR-NEXT 19
Time delay 23
STEP 27
Review 28

CHAPTER 4: A PICTURE IS WORTH 1000 WORDS

GR.5 29
COLOR. 29
PLOT H,V 29
GR.Ø 31
DRAWTO H,V 42
Review 51

Creative Learning Association
Charleston, IL 61920

Table of Contents (continued)

CHAPTER 5: MATH IS A PIECE OF CAKE!

Page

Math Operations 52
Review 56
Reference 57
Math Reference. 60

Creative Learning Association
Charleston, IL 61920

CHAPTER 1: INPUT IS OFTEN BETTER THAN OUTPUT

INFORMATION

The command INPUT is used to add information to a program that is running. It allows one to put in information to be used in the program. By using INPUT, a program can be run many times with different INPUTed information thus producing different results.

ENTER/RUN/OBSERVE

```
10 INPUT X
20 PRINT "YOUR NUMBER IS ";X
```

When you see a question mark on the screen, type a number and press [RETURN]. When the computer comes to an INPUT statement, it waits for the information. When it receives it, it goes on to the next step in the program. When you use INPUT, always include a letter to represent the information that will be INPUTed. Since these letters can have different values at different times, they will be called variables.

CHANGE IT/RUN/OBSERVE

```
10 PRINT "PLEASE ENTER A NUMBER";
20 INPUT N
30 GR.0
40 PRINT "YOU ENTERED THE NUMBER ";N
```

You may use GR.Ø after an INPUT statement to clear the screen before your message that uses the INPUT is printed.

CHALLENGE

Write a program that asks for your age in years and then prints a sentence using that number on the screen. Make it a very complimentary message . . . after all, you're the boss!

ANOTHER CHALLENGE

Write a program that asks how many hours you worked, but have the INPUTed value appear on the line below the question. Then print a sentence using that number on the screen.

INFORMATION

You can INPUT letters and words by using a special type of variable called a string variable. A string variable has a dollar sign after it. Use string variables whenever the INPUTed information is anything other than just numbers. For example, for INPUT that consists of letters or symbols along with numbers, such as 37%, you will need to use a string variable.

ENTER/RUN/OBSERVE

```
10 PRINT "WHAT IS YOUR NAME";
20 INPUT N$
30 PRINT "HELLO ";N$;"!"
```

NOTE: N$ is pronounced "N string".

When you INPUT a value, you get ERROR - 9 at line 20. You can prevent the error by adding this line.

```
 5 DIM N$(20)
```

INFORMATION

DIM N$(S) sets aside space in the computer's memory. The value of S determines how much space is reserved for N$.

ENTER/RUN/OBSERVE

```
 5 GR.Ø
1Ø DIM N$(5)
2Ø PRINT "TYPE IN THE NAME OF A PET"
3Ø INPUT N$
4Ø PRINT
5Ø PRINT "HELLO ";N$
```

RUN and type in a 4 letter animal.

RUN again with a 5 letter animal.

RUN again with a 6 letter (or more) animal.

Notice that up to 5 letters are accepted and used because that is all the space that was reserved for N$ by line 1Ø.

CHALLENGE

Create a program that asks you to INPUT your favorite movie star. Then have it PRINT a line using that star's name.

ENTER/RUN/OBSERVE

```
 1Ø GR.Ø
 2Ø DIM N$(2Ø)
 3Ø DIM T$(2Ø)
 4Ø DIM P$(9)
 5Ø PRINT "WHAT IS YOUR NAME?"
 6Ø INPUT N$
 7Ø PRINT "WHAT TOWN DO YOU LIVE IN?"
 8Ø INPUT T$
 9Ø PRINT "WHAT IS YOUR TELEPHONE NUMBER?"
100 INPUT P$
110 GR.Ø
120 PRINT
130 PRINT "IF ANYONE WANTS TO TALK TO SOMEONE"
140 PRINT
150 PRINT "WHO LIVES IN ";T$;","
160 PRINT
170 PRINT "CALL ";P$;" AND YOU CAN SPEAK"
180 PRINT
190 PRINT "TO ";N$;"."
```

Notice how each string was DIMensioned before the string was used. Otherwise an error message results.

CHALLENGE

Create a similar program that tells what might happen to someone in your ZIP CODE area.

INFORMATION

String variables and numeric variables can be used in the same program.

ENTER/RUN/OBSERVE

```
10 GR.0
20 DIM Y$(10)
30 DIM M$(10)
40 DIM Z$(12)
50 DIM N$(20)
60 X=10
70 Y$=" CAN EAT "
80 M$=" MINUTES."
90 Z$=" PIZZAS IN "
100 PRINT "PLEASE ENTER A NAME ";
110 INPUT N$
120 PRINT "PLEASE ENTER A NUMBER ";
130 INPUT N
140 PRINT N$;Y$;N;Z$;X;M$
```

You can assign values to variables directly or you may enter them as INPUT. The variable's name must be on the left, and the value of the variable on the right. The values of all string variables must be enclosed in quotation marks.

Notice that only the string variables needed DIM statements.

CHALLENGE

Change the message in the "pizza" program by changing line 14Ø.

ENTER/RUN/OBSERVE

```
10 GR.0
20 DIM A$(15)
40 DIM N$(15)
50 DIM S$(15)
60 PRINT "NAME A COLOR ";
70 INPUT A$
80 PRINT "NAME A STATE "
90 INPUT S$
120 PRINT "ENTER A PLURAL NOUN "
130 INPUT N$
140 PRINT "ENTER A NUMBER ";
150 INPUT N
160 GR.0
170 PRINT "ONE DAY ";N;" ";A$;" ";N$;" ";
"WERE IN THE PARK IN ";S$;"."
```

RUN the above program with different INPUTed values.

CHALLENGE

Write a program to request a name, a time, and a place. Print out an invitation to the name you INPUT for a secret rendezvous (meeting).

ANOTHER CHALLENGE

Create an ad for a refreshment stand that includes both numeric and string variables and allows you to INPUT changes.

INFORMATION

Normally the computer prints in light letters on a dark background. To switch this, the ATARI LOGO KEY, [ATARI LOGO KEY] , is used. Pressed again, it will return the screen to normal.

ENTER/RUN/OBSERVE

```
10 GR.0
20 PRINT "[ATARI LOGO KEY] DARK ON LIGHT [ATARI LOGO KEY]"
30 GOTO 20
```

CHANGE IT/RUN/OBSERVE

Break your program and add:

```
25 PRINT "LIGHT ON DARK"
```

RUN your program again. This time press the [CTRL] and [1] keys at the same time.

To go on, press [CTRL] [1] again. Notice READY didn't appear.

Break your program and clear memory by typing NEW . Clear the monitor screen by holding down the [SHIFT] key and pressing the [CLEAR <] key.

ENTER/RUN/OBSERVE

Press [ʃʃ], the reverse video key.

```
**********************************************
*                                            *
* 1Ø PRINT                                   *
*                                            *
**********************************************
```

Notice: You get an error message when commands and line numbers are written in reverse video.

CHANGE IT/RUN/OBSERVE

Press [ʃʃ] to return to normal printing.

```
**********************************************
*                                            *
* 1Ø PRINT [ʃʃ] "ATARI"                      *
*                                            *
**********************************************
```

Notice: You get an error message if quotation marks are written in reverse video.

CHANGE IT/RUN/OBSERVE

Press [ʃʃ].

```
**********************************************
*                                            *
* 1Ø PRINT " [ʃʃ] ATARI [ʃʃ] "               *
*                                            *
**********************************************
```

Only information within the quotation marks should be written in reverse video.

CHALLENGE

Create a birthday message for a friend using the ATARI LOGO key. Make it a humorous one.

REVIEW: Choose the command or key which will cause the function described below.

COMMAND/KEY	FUNCTION
	This command allows information to be put into a program while the program is running.
	This symbol must be included after a variable name if the variable is a string variable.
	This command sets aside memory space for string variables.
	This key is used to change the computer output to dark printing on a light background.
	Pressing these keys causes a program to stop without READY appearing. Pressing them again causes the program to continue.

CHAPTER 2: UP, UP AND ARRAY!

INFORMATION

Arrays are a convenient way to store information. You can think of them as lots of little boxes in the computer's memory into which you can place information. There are several types of arrays. For the time being, you will just be working with single arrays.

A single array is somewhat like the mailboxes in an apartment building.

-	-	-	-	-	-	-	-	-
Ø	1	2	3	4	5	6	7	8

There is one mailbox for each apartment. Each box has a label.

Suppose an array has four numbered boxes. The following program will set the array up and number each box. The array is called B. The number for the DIMension of this array is 3.

```
DIM B(3).
```

ENTER/RUN/OBSERVE

```
 5 DIM B(3)
10 B(1)=154
20 B(2)=325
30 B(3)=517
40 GR.0
50 PRINT "BOX ONE OF THIS ARRAY CONTAINS ";
   B(1)
60 PRINT
70 PRINT "BOX TWO OF THIS ARRAY CONTAINS ";
   B(2)
80 PRINT
90 PRINT "BOX THREE OF THIS ARRAY CONTAINS ";
   B(3)
```

Notice the number 3 used in the DIM command is the largest numbered box in the array.

CHANGE IT/RUN/OBSERVE

Change the value stored in B(2) to 438 and RUN again.

CHANGE IT/RUN/OBSERVE

```
*******************************************
* 3Ø B(Ø)=517                             *
*                                         *
*1ØØ PRINT "BOX ZERO OF THIS ARRAY CONTAINS*
*";B(Ø)                                   *
*******************************************
```

The first box in an array is labeled zero not one. The dimension statement is one less than the actual number of boxes used if box #Ø is used.

CHALLENGE

Set up an array that has four mailboxes. Store a number in each. Then have your program print out the numbers stored in each box.

INFORMATION

The command REM shows up in the listing of a program, but not when the program is run. When you LIST, REM helps to make a program easier to understand by REMinding the programmer of what the program lines do and REMarking about something you want to REMember.

ENTER/RUN/OBSERVE

```
10 REM  ***LINE 20 SETS ASIDE 11 SPACES
FOR ARRAY A.***

20 DIM A(10)

30 REM ***LINE 40 CLEARS THE SCREEN***

40 GR.0

50 PRINT "PLEASE ENTER A NUMBER."

60 INPUT N

70 REM ***STORE N IN THE ARRAY***

80 A(X)=N

90 PRINT "# ";X;" CONTAINS ";A(X)

100 REM ***COUNT HOW MANY NUMBERS ARE
STORED***

110 X=X+1

120 GOTO 50
```

Run this program, then look carefully at the listing. The REM statements appear in the LISTing, but they have no effect on the RUN.

How many numbers can you enter? (Try at least 15!)

ENTER/RUN/OBSERVE

```
*****************************************
* 1Ø REM ***NANCY P. TAITT***           *
* 2Ø REM ***Ø3/Ø4/83***                 *
* 3Ø REM ***HOUSEHOLD BUDGET PROGRAM*** *
* 4Ø PRINT "I'M BROKE!"                 *
*****************************************
```

It is not necessary to always use asterisks (*) as shown here, but they set the REM statements off nicely in a program and make them easier to spot and read.

CHALLENGE

Create a program using INPUT and arrays. Use REM statements to document it at the beginning, and also use some throughout to explain your program.

REVIEW: Choose the command or key which will cause the function described below.

COMMAND/KEY	FUNCTION
	This is a series of locations in the computer's memory into which you can place information.
	This command is used to include remarks in a program's listing.
	This command is used to set aside memory locations to be used for an array.

CHAPTER 3: FOR YOUR NEXT PROGRAM, TRY THIS!

INFORMATION

FOR-NEXT enables the computer to do something a certain number of times and then go on to the rest of the program.

ENTER/RUN/OBSERVE

```
 5 REM ** THIS WILL PRINT FROM 1 TO 1Ø **
1Ø FOR X=1 TO 1Ø
2Ø PRINT X
3Ø NEXT X
```

CHANGE IT/RUN/OBSERVE

```
2Ø PRINT "X"
```

Notice the effect of quotation marks. A FOR-NEXT loop includes the FOR line, the NEXT line, and all the lines in between.

CHALLENGE

Change this program to make the computer print numbers from 5 to 25.

ENTER/RUN/OBSERVE

```
10 GR.0
20 PRINT "PLEASE ENTER A NUMBER";
30 INPUT N
40 FOR X=1 TO N
50 PRINT X
60 NEXT X
```

RUN this several times with different INPUTed values for N.

CHANGE IT/RUN/OBSERVE

Change lines 20 and 40.

```
20 PRINT "ENTER A NUMBER LESS THAN 100 "
40 FOR X=N TO 100
```

LIST your program. Study it carefully. RUN several times again with different INPUTed values for N.

CHALLENGE

Create a program that will count from the first number you INPUT to the second number of INPUT.

ENTER/RUN/OBSERVE

```
  2 REM ***LINE 5 RESERVES 7 ARRAY SPACES***
  5 DIM B(6)
 1Ø GR.Ø
 2Ø REM ***INPUT 6 NUMBERS***
 3Ø FOR X=1 TO 6
 4Ø PRINT "PLEASE ENTER A NUMBER "
 5Ø INPUT N
 53 REM ***STORE N IN THE ARRAY***
 55 B(X)=N
 6Ø NEXT X
 7Ø GR.Ø
 8Ø PRINT
 9Ø PRINT "HERE ARE YOUR NUMBERS"
1ØØ REM ***PRINT THE ARRAY***
11Ø FOR X=1 TO 6
12Ø PRINT B(X)
13Ø NEXT X
```

Notice that there are two FOR-NEXT loops in this program. The first one (lines 3Ø-6Ø) stores the INPUTed numbers in array B. The second (lines 11Ø-13Ø) prints the numbers stored in array B.

Whenever you use a FOR, there must be a corresponding NEXT somewhere later in the program. The variable used in the NEXT statement must be the same as that used in the FOR statement.

CHANGE IT/RUN/OBSERVE

Add the following lines to your program:

```
140 PRINT
145 S=0
150 REM ***THIS ADDS THE NUMBERS STORED IN
THE ARRAY B***
160 FOR X=1 TO 6
170 S=S+B(X)
180 NEXT X
190 PRINT "THE SUM OF YOUR NUMBERS IS ";S
```

CHANGE IT/RUN/OBSERVE

Add these lines:

```
200 PRINT
210 PRINT "THE AVERAGE OF YOUR NUMBERS IS ";
S/6
```

NOTE: / is the division sign on your computer.

CHALLENGE

Change the previous program to:

1. Accept 8 numbers.
2. Print all 8 numbers.
3. Print their sum.
4. Print their average.

CHANGE IT/RUN/OBSERVE

Add these lines to your current program:

```
125 FOR T=1 TO 500
128 NEXT T
```

There are no commands between the FOR and NEXT. The computer performs the loop 500 times and then goes on. This is called a time delay.

RUN this program again and notice the delay before each word is printed.

CHANGE IT/RUN/OBSERVE

Make this change.

```
125 FOR T=1 TO 500: NEXT T
128
```

The colon allows you to put two different commands on the same line.

CHALLENGE

Change the numbers in line 125 so the words will print more slowly.

INFORMATION

GOTO repeats an operation until you make it stop by pressing CTRL 1 or BREAK. With FOR-NEXT, the programmer controls how many times something happens. The number of times can either be written in the program, or INPUTed while the program is running.

ENTER/RUN/OBSERVE

```
10 GR.0
20 PRINT
30 PRINT "HOW HIGH WOULD YOU LIKE THE COMPU
TER TO COUNT ";
35 INPUT X
40 FOR N=1 TO X
50 PRINT N
55 REM ***LINES 60-70 ARE A TIME DELAY
LOOP***
60 FOR T=1 TO 500
70 NEXT T
80 NEXT N
```

Notice that when you have one FOR-NEXT loop inside of the other (as in lines 40-80 above)

one loop must be completely inside of the other. In other words, the T FOR-NEXT loop starts and ends inside of the N loop. Try running the program with lines 7Ø and 8Ø switched and see what happens.

CHALLENGE

Change the last program so that:

1. The time delay between numbers is shorter.
2. The time delay is longer.
3. The time delay is one second long.

INFORMATION

Unless you tell it otherwise, the computer takes steps of +1 in FOR-NEXT loops. But you can step by 5, 27, -3, -1ØØ, or any other number by using the STEP command.

ENTER/RUN/OBSERVE

```
1Ø GR.Ø
2Ø FOR X=Ø TO 25 STEP 5
3Ø PRINT X
4Ø NEXT X
5Ø FOR X=25 TO Ø STEP -5
6Ø PRINT X
7Ø NEXT X
```

CHALLENGE

Change the program so that it will print the multiples of 3 from Ø to 333. Use time delays if you wish to slow down the printing and make the result more attractive.

REVIEW: Choose the command or key which will cause the function described below.

COMMAND/KEY	FUNCTION
	This is a command used to control how many times something happens.
	Used with FOR-NEXT, this causes steps other than +1 to be taken.
	This symbol allows you to put two different commands on the same line.

CHAPTER 4: A PICTURE IS WORTH 1000 WORDS

INFORMATION

Sometimes words alone are not enough. A picture is needed. The command PLOT H,V causes a small rectangular section of the screen to light. The first value (H) indicates the horizontal position on the screen while the second value (V) indicates the vertical position. Graphics and pictures can be created by controlling which rectangles are lit.

ENTER/RUN/OBSERVE

```
10 GR.5
20 COLOR 1
30 PLOT 20,10
```

Notice where the lighted rectangle appears.

Line 10 puts the computer in the GRAPHICS 5 mode.

Line 20 selects the COLOR if you have a color monitor. (If not, it selects the type of pattern that will appear at H,V.) Color 1 is red.

Line 3Ø causes the small rectangle 2Ø units over from the left side of the screen and 1Ø units down from the top to be lit. (H=2Ø, V=1Ø) The three commands---GRAPHICS, COLOR, and PLOT are necessary in order to light the small rectangles on the screen.

In GRAPHICS mode 5, the horizontal value can take any number from Ø to 79; the vertical value can be a number from Ø to 39.

CHANGE IT/RUN/OBSERVE

```
3Ø PLOT 5Ø,1Ø
```

CHANGE IT/RUN/OBSERVE

```
3Ø PLOT 79,39
```

INFORMATION

LIST your program. Notice that when you are in the GR.5 mode only the last line of the program is shown at the bottom of the screen, in the TEXT WINDOW. Actually there is room for four lines in the TEXT WINDOW, but READY takes up the bottom three lines. The computer is still in the GRAPHICS 5 mode, from line 1Ø. To return to the full screen, press [SYSTEM RESET] or type GR.Ø and press [RETURN].

Now LIST your program.

CHANGE IT/RUN/OBSERVE

Add these lines:

```
*****************************************
* 1Ø PRINT "HI"                         *
*                                       *
* 4Ø PLOT 3Ø,1Ø                         *
*                                       *
*****************************************
```

RUN your program.

Adding line 1Ø replaces the graphics command, putting you back in the TEXT mode. When you run your program, it tries to do graphics in the TEXT mode and gives you an error message.

This reminds you that you must be in the

GRAPHICS mode in order to use the PLOT command.

CHANGE IT/RUN/OBSERVE

```
10 GR.5
```

Now you are back in the GRAPHICS 5 mode with

the TEXT WINDOW at the bottom.

ENTER/RUN/OBSERVE

```
10 GR.5
20 COLOR 2
30 PLOT 55,20
40 PRINT "COLOR 2 IS YELLOW"
```

Notice where the lighted rectangle appears.

CHANGE IT/RUN/OBSERVE

Press [SYSTEM RESET] to restore the full screen and

add these lines.

```
20 COLOR 3
30 PLOT 0,0
40 PRINT "COLOR 3 IS GREEN"
```

Notice where the lighted rectangle appears. This position is called home because it is where the cursor goes when the screen is cleared. It is also the position from which H, the horizontal value, and V, the vertical value, are counted.

CHALLENGE

Write a program to cause all four corners of the screen to be lit.

ENTER/RUN/OBSERVE

```
10 GR.5
20 COLOR 1
30 PLOT 80,4
```

The error message occurs because 79 is the largest horizontal number allowed in the GRAPHICS 5 mode. Although there are 80 spaces across the screen, they are numbered 0 to 79.

CHANGE IT/RUN/OBSERVE

```
**********************************************
* 3Ø PLOT 2Ø,48                              *
**********************************************
```

The computer cannot light the 48th space down!

CHANGE IT/RUN/OBSERVE

```
**********************************************
* 3Ø PLOT 2Ø,42                              *
**********************************************
```

What happened? Although in the GRAPHICS 5 mode the graphics screen is 4Ø units long (Ø-39), the text window at the bottom adds an additional 8 units. An error message does not occur until Y, the vertical value, is greater than 47.

ENTER/RUN/OBSERVE

```
**********************************************
* NEW                                        *
**********************************************
```

Now enter this:

```
10 GR.5
20 COLOR 2
30 PLOT 35,16
40 PLOT 45,16
50 PLOT 40,20
60 PLOT 35,24
70 PLOT 45,24
80 PRINT "FIVE ON THE FACE OF A DIE"
```

CHANGE IT/RUN/OBSERVE

```
50
80 PRINT "FOUR ON THE FACE OF A DIE"
```

CHALLENGE

Write a program to make the face of a die with two spots. Print a message in the text window.

ENTER/RUN/OBSERVE

```
NEW
```

Now enter this:

```
10 GR.5
20 COLOR 2
30 PLOT 10,15
40 PLOT 10,16
50 PLOT 10,17
60 PLOT 10,18
70 PLOT 10,19
80 PLOT 10,20
```

CHANGE IT/RUN/OBSERVE

```
30 FOR V=15 TO 20
40 PLOT 10,V
50 NEXT V
60
70
80
```

Since the H and V in the PLOT command are variables, a line can be drawn using a FOR-NEXT loop without having to PLOT each point. Think through the above program giving V each value from 15 to 20.

CHALLENGE

Extend the line to the bottom of the graphics screen; to the top of the graphics screen.

Move the line to the center of the screen.

ENTER/RUN/OBSERVE

First clear the memory with NEW.

```
10 GR.5
20 COLOR 3
30 FOR X=10 TO 40
40 PLOT X,20
50 NEXT X
```

CHANGE IT/RUN/OBSERVE

```
30 FOR X=10 TO 60
```

CHALLENGE

Extend the line to the left edge of the screen.

Move the line down 10 units.

ANOTHER CHALLENGE

Make a large H on the screen using a FOR-NEXT loop with the PLOT command. Use the graph paper on the next page to plan.

Atari Video Display Worksheet

ENTER/RUN/OBSERVE

Clear the old program from memory, then enter:

```
5 GR.5
1Ø FOR X=1 TO 6Ø
2Ø COLOR 1
3Ø PLOT X,1Ø
4Ø NEXT X
```

CHANGE IT/RUN/OBSERVE

```
35 COLOR 4
36 PLOT 3Ø,1Ø
```

Lighting a rectangle to match the background (COLOR 4 is black) is useful in making designs.

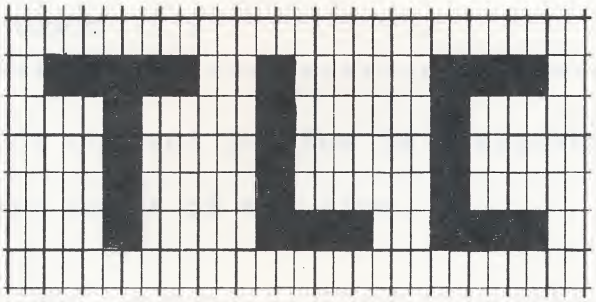
CHANGE IT/RUN/OBSERVE

```
36 PLOT X-1,1Ø
```

This will turn off the rectangle one step behind the value of X. RUN and LIST this program several times as you study it carefully.

CHALLENGE

Using FOR-NEXT commands, create a program that will draw this on the screen.



Can you make it blink on and off? In color?

INFORMATION

The ATARI has another method of drawing lines.

ENTER/RUN/OBSERVE

```
10 GR.5
20 COLOR 3
30 PLOT 0,0
40 DRAWTO 79,0
```

The PLOT command in line 30 designates the starting point of the line.

The DRAWTO command in line 40 instructs the cursor to draw a line to the spot 80 spaces across (H=79) but not down any, since V=0.

CHANGE IT/RUN/OBSERVE

Press [SYSTEM RESET] and LIST your program.

Add this:

```
50 DRAWTO 79,39
```

No additional PLOT command was necessary as the line was drawn from the last position of the cursor (H=79,V=Ø).

CHALLENGE

Add to the above program so a border is drawn around the edge of the screen.

ENTER/RUN/OBSERVE

Type NEW.

```
10 GR.5
20 COLOR 1
30 PLOT 30,12
40 DRAWTO 50,12
50 DRAWTO 50,24
60 DRAWTO 30,24
70 DRAWTO 30,12
```

CHALLENGE

Draw a square somewhere else on the screen using the DRAWTO commands.

ENTER/RUN/OBSERVE

```
10 GR.5
20 COLOR 1
30 PLOT 30,10
40 DRAWTO 30,29
50 PLOT 50,10
60 DRAWTO 50,29
65 COLOR 1
70 PLOT 30,15
80 DRAWTO 50,15
90 FOR T=1 TO 500: NEXT T
100 COLOR 4
110 PLOT 31,15
120 DRAWTO 49,15
130 FOR T=1 TO 500: NEXT T
140 GOTO 65
```

Lines 20-80 draw the goal posts and crossbar in red. Lines 100-120 draw the crossbar in black, causing it to disappear. Line 140 causes the red crossbar to be drawn again, thus producing the blinking effect. The timing loops in lines 90 and 130 slow things down so the blinking can be seen. Study this program carefully.

CHALLENGE

Draw a design of your choosing on the screen.

Make it blink.

INFORMATION

Areas can be painted on the screen using the

FOR-NEXT loop with the DRAWTO command.

ENTER/RUN/OBSERVE

To fill a whole screen, try:

```
10 GR.5
20 COLOR 1
30 FOR X=0 TO 79
40 PLOT X,0
50 DRAWTO X,39
60 NEXT X
```

Think through several values of X to under-

stand how this works.

CHANGE IT/RUN/OBSERVE

```
30 FOR X=0 TO 39
```

CHALLENGE

Can you paint just the left fourth of the screen yellow?

Can you paint just the right half of the screen green?

ENTER/RUN/OBSERVE

```
10 GR.5
20 COLOR 1
30 FOR X=0 TO 30
40 FOR Y=0 TO 15
50 PLOT X,Y
60 NEXT Y
70 NEXT X
80 PRINT "COLOR 1 IS RED"
```

RUN again and LIST.

CHANGE IT/RUN/OBSERVE

Add these lines:

```
90 COLOR 2
100 FOR X=31 TO 60
```

(program continued)

```
110 PLOT X,0
120 DRAWTO X,15
130 NEXT X
140 PRINT "COLOR 2 IS YELLOW"
```

Run again and observe which shape is made more quickly.

It is faster to use the DRAWTO command than to use the FOR-NEXT loops. Study this program carefully.

CHALLENGE

Add to the above program to make a green rectangle appear to the right of the yellow one.

ENTER/RUN/OBSERVE

Clear the memory of your computer before starting this program.

```
  5 REM *** PAINTS THE STAIRCASE***
10 GR.5
20 COLOR 2
30 FOR X=0 TO 79
```

(program continued)

```
 4Ø PLOT Ø,Ø
 5Ø DRAWTO X,39
 6Ø NEXT X
```

CHANGE IT/RUN/OBSERVE

Add these lines:

```
 65 REM ***PAINTS THE SKY***
 7Ø COLOR 3
 8Ø FOR X=79 TO 1 STEP -1
 9Ø PLOT 79,39
1ØØ DRAWTO X,Ø
11Ø NEXT X
```

CHANGE IT/RUN/OBSERVE

Add these lines:

```
115 REM ***MAKES FIGURE***
12Ø COLOR 1:PLOT 18,6
13Ø COLOR 4:PLOT 18,7
14Ø COLOR 1:PLOT 18,8
```

CHANGE IT/RUN/OBSERVE

Add these lines:

```
118 REM ***MAKES FIGURE MOVE***
120 Y=6
130 FOR X=18 TO 79 STEP 2
140 COLOR 1:PLOT X,Y
150 COLOR 4:PLOT X,Y+1
160 COLOR 1:PLOT X,Y+2
170 Y=Y+1
180 NEXT X
190 END
```

CHANGE IT/RUN/OBSERVE

Add these lines:

```
164 REM ***ERASES TRAIL***
165 COLOR 3:PLOT X,Y
166 COLOR 3:PLOT X,Y+1
167 COLOR 3:Plot X,Y+2
```

The commands (165-167) cause the figure to be erased after each time it is plotted, because color 3 is the background color. Think through several values of X to understand how it works.

CHALLENGE

Insert a time delay loop between lines 16Ø and 167 to slow the figure going down the stairs.

ANOTHER CHALLENGE

Change the color of the bottom staircase.

Change the color of the sky.

Change the color of the little man.

SUPER CHALLENGE

Make the man go back up the stairs. Experiment with different starting points for the staircase by changing line 4Ø. Be creative!

REVIEW: Choose the command or key which will cause the function described below.

COMMAND/KEY	FUNCTION
	Use this command to produce four colors or patterns on your monitor screen in graphics 5.
	This command lights one small rectangle on the screen.
	This command places the computer in the graphics 5 mode (80 x 40) and produces a split screen.
	Use this command to draw a line from the cursor's present position to H,V.

CHAPTER 5: MATH IS A PIECE OF CAKE!

INFORMATION

The computer you are commanding is also very good at mathematics. But you must tell it what to do. Below are some of the mathematical operations it can do. Try each of the examples in the direct mode, without giving them line numbers.

OPERATION	SYMBOL	EXAMPLE
Add	+	PRINT 2+2 RETURN
Subtract	-	PRINT 4-2 RETURN
Multiply	*	PRINT 4*2 RETURN
Divide	/	PRINT 4/2 RETURN

ENTER/RUN/OBSERVE

Try entering this.

```
*****************************************
* PRINT 3*4+2                           *
*                                       *
*****************************************
```

Is it 3 times 6 or 12 plus 2? Try this,

```
*****************************************
* PRINT 3*(4+2)                         *
*                                       *
*****************************************
```

and this,

```
PRINT 2*3+2
```

and this.

```
PRINT 2*(3+2)
```

CHANGE IT/RUN/OBSERVE

It can also do very complex things if you show it how!

```
10 GR.0
20 INPUT X
30 W=(2*X)-3+(4*X)/(6+X)
40 PRINT X,W
50 GOTO 20
```

CHALLENGE

What is the largest value of X this program will accept. (Be lazy.)

CHANGE IT/RUN/OBSERVE

This can be very useful for creating tables.

```
10 GR.0
20 FOR X=1 TO 5
30 PRINT X,X*2,(2*X)/3
40 NEXT X
```

Don't worry if these do not appear in nice straight columns. You will learn to take care of that later.

CHALLENGE

Create a program that will produce the following table:

	2	3	4	5	6
1	2	3	4	5	6
2	4	6	8	10	12
3	6	9	12	15	18
4	8	12	16	20	24
5	10	15	20	25	30
6	12	18	24	30	36

CHALLENGE

In the United States, we use dollars as our unit of money. In Ecuador, South America, the sucre is used, in Britain it is the pound, in Japan the yen, in West Germany the mark, and in India the rupee. Suppose on January 18, 1983 the following conversions are true:

33	sucres	= 1 U.S. dollar
.63	pound	= 1$
232	yen	= 1$
2.40	marks	= 1$
9.8	rupees	= 1$

Write a program that asks you to input an amount in dollars, and will then print the equivalent amount of money in sucres, pounds, yen, marks, and rupees, along with the name of the country where that type of money is used.

REVIEW: Choose the command or key which will cause the function described below.

SYMBOL	FUNCTION
	Performs addition in BASIC.
	Performs subtraction in BASIC.
	Performs multiplication in BASIC.
	Performs division in BASIC.

REFERENCE

COMMAND/KEY	FUNCTION
ARRAY	This is a series of locations in the computer's memory into which you can place information.
ATARI LOGO KEY	This is used to change the computer output to dark printing on a light background.
(Colon) :	This symbol allows you to put two different commands on the same line.
COLOR	Use this command to produce four colors or patterns on your monitor screen in graphics 5.
CTRL 1	Pressing these keys causes a program to stop without READY appearing. Pressing them again causes the program to continue.
DIM X	This command is used to set aside memory locations to be used for an array.
DIM X$	This command sets aside memory space for string variables.

REFERENCE (continued)

COMMAND/KEY	FUNCTION
DRAWTO H,V	Use this command to draw a line from the cursor's present position to H,V.
FOR-NEXT	This command is used to control how many times something happens.
GR.Ø	This command places the computer in the text mode and clears the screen.
GR.5	This command places the computer in the graphics 5 mode (80 x 40) and produces a split screen.
INPUT	This command allows information to be inserted into a program while the program is running.
PLOT H,V	This command lights one small rectangle on the screen.
REM	This command is used to include remarks in a program's listing.
STEP	Used with FOR-NEXT, this causes increments other than +1 to be taken.

REFERENCE (continued)

COMMAND/KEY	FUNCTION
SYSTEM RESET	This key when pressed changes from a split screen (graphics 5 mode) to a full screen (text mode, graphics Ø).

MATH REFERENCE

SYMBOL	FUNCTION
+	Performs addition in BASIC.
-	Performs subtraction in BASIC.
*	Performs multiplication in BASIC.
/	Performs division in BASIC.

CONGRATULATIONS!!!

You have now finished the orange instructional level of TLC FOR GROWING MINDS and have a working vocabulary of BASIC commands you can now use to command your microcomputer.

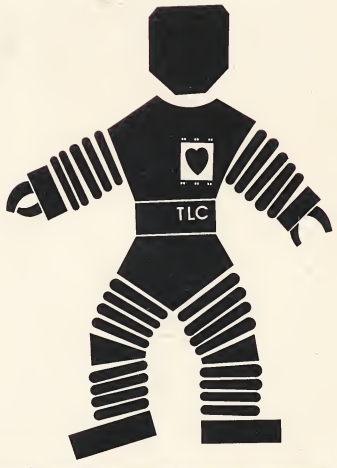
You are now ready to do TWO things:

(1) Gain experience using the commands you have already learned in a variety of new ways, and

(2) Learn more commands.

The TLC Project Book, Volume I/II, will help you gain experience using the commands you have already learned, and Volume III (Yellow) of TLC FOR GROWING MINDS will help you learn more commands.

Creative Learning Association
Charleston, IL 61920



THINKING · LEARNING · CREATING

TLC

for Growing Minds

CREATIVE LEARNING ASSOCIATION, INC.
RR#4, Box 330
Charleston, Illinois 61920
(217) 345-1010

ATA-R-2

ISBN 0-88193-072-5